**Sokoban**

**1. Introductie**



Menu spel (versie 1; alléén toetsenbord besturing)

Dit is een simpele versie van Sokoban, gemaakt met Rapid Q, met tien puzzels ingebouwd; met het ontwerpprogramma kunnen heel eenvoudig zelf puzzels gemaakt worden. Om met die zelfgemaakte puzzels een eigen versie van het spel te maken is een minieme hoeveelheid werk nodig.

Om het spel te spelen is alleen het **Sokoban\_Nl.EXE** bestand nodig.

Om puzzels te ontwerpen is alleen het bestand **SokobanBuilder.EXE** nodig.

Om met de eigen gemaakte puzzels een eigen editie van het spel te maken is ten minste de Rapid Q omgeving vereist met daarbij de materialen (verschillende kleine grafische elementen, broncode) die in het programma gebruikt worden.

Alle benodigde materialen zijn in het **Sokoban.ZIP** bestand opgenomen (zonder de genoemde EXE bestanden), voor Rapid Q is een set van drie ZIP bestanden beschikbaar (waarvan er maar één direct vereist is voor dit doel; **RQ\_beta.ZIP**).

In deze tekst wordt stap voor stap uitgelegd hoe je een eigen versie (met eigen puzzels) kunt maken.

**2. Voorbereiding**

Maak een *directory* aan (bij voorkeur **C:\Sokoban**) en kopieer daar de vier **ZIP** bestanden naar toe, pak vervolgens **Sokoban.ZIP** uit in deze map.

Maak een directory voor Rapid Q aan en pak de drie RQ ZIP bestanden daarheen uit. Ga naar die map en maak een desktop snelkoppeling voor **RAPIDQ.EXE** en voor **contents.HTML** (index Rapid Q gebruiksaanwijzing)

**3. Puzzels maken**

De eerste vereiste voor het maken van een eigen versie is het maken van eigen puzzels, daarvoor is het hiernaast getoonde programma bedoeld.

Trek de elementen die je wilt gebruiken op de verdieping; let er op dat je één magazijnmedewerker plaatst en even veel doelen als kisten (bij opslaan controleert het programma dit).

Het programma slaat iedere puzzel op in de vorm van drie bestanden;

* SBL voor het ontwerp programma
* BMP een plaatje van het ontwerp
* TXT in de vorm van een stukje tekst

Bij het opslaan moet een naam gekozen worden, gebruik daarvoor één woord; alleen letters en cijfers, geen spaties of bijzondere tekens.

Die tekst (het TXT bestand) ziet er voor ons voorbeeld als volgt uit;

 data 1,1,5,5,5,5,5,5,5,5,5,5,1,1, 1,6,1,1,1,1,1,1,1,1,1,1,6,1, 5,1,6,6,1,1,1,1,1,1,6,6,1,5, 5,1,6,5,1,1,1,1,1,1,5,6,1,5

 data 5,1,1,1,1,1,2,1,2,1,1,1,1,5, 5,1,1,1,1,2,1,2,1,2,1,1,1,5, 5,1,1,1,2,1,2,1,2,1,1,1,1,5, 5,1,1,1,1,2,1,2,1,2,1,1,1,5

 data 5,1,1,1,2,1,2,1,2,1,1,1,1,5, 5,1,1,1,1,2,1,2,1,1,1,1,1,5, 5,1,6,5,1,1,1,1,1,1,5,6,1,5, 5,1,6,6,1,1,3,1,1,1,6,6,1,5

 data 1,6,1,1,1,1,1,1,1,1,1,1,6,1, 1,1,5,5,5,5,5,5,5,5,5,5,1,1, Vrijbaan

Het is deze tekst die de *broncode* van het spel geplakt moet worden. De naam van de ‘verdieping’ in deze tekst kan bijgewerkt worden, dat is hoe de titel in het spel zal verschijnen.

**4. Een eigen Sokoban versie maken met Rapid Q**



Start Rapid Q (met de snelkoppeling) en accepteer daarbij de waarschuwingen (Windows is veranderd, Rapid Q niet; dat veroorzaakt die waarschuwingen). Je ziet dan iets als hierboven verschijnen.

 

Klik **hier** (op het rood omlijnde symbool in het linker plaatje) en navigeer naar de map met de Sokoban spullen, kies daar het pictogram dat je wilt gebruiken (voorbeeld plaatje rechts). **Let op;** er kan alleen een pictogram van 32 bij 32 pixels met 16 kleuren gebruikt worden.

 

Het gekozen pictogram verschijnt in het ontwerp venster en op de plaats van het basis pictogram, klik nu **hier** om het code venster te maximaliseren. Ga nu naar **File** – **Open…** (of klik het map icoon) en kies **Sokoban\_Nl.BAS**

Ga (schuifbalk rechts) naar het einde van de broncode en je ziet de puzzel ontwerpen staan.

Open een eigen puzzel ontwerp (zie pagina 2, gebruik Notepad o.i.d.) TXT bestand, selecteer de inhoud (druk **Ctrl** + **A**) en buffer deze (druk **Ctrl** + **C**).

Wissel naar Rapid Q, ga (schuifbalk rechts) naar het einde van de broncode en je ziet de puzzel ontwerpen staan. Selecteer één van de ontwerpen in de broncode;



Zet nu de buffer (druk **Ctrl** + **C**) in de broncode, het oude ontwerp wordt daarbij overschreven;



Kies nu **Run** – **run** (of druk **F5**) om van de broncode een programma te maken; kies het nummer van de zojuist vervangen puzzel.



Dit is in principe voldoende om een eigen versie te maken. Om je eigen versie duidelijk herkenbaar te maken is het natuurlijk leuker om nog wat dingetjes aan te passen; het programma is er op ingericht je in staat te stellen de kleur van de ‘lucht’ simpel aan te passen.

Ga naar het begin van de broncode en kijk naar deze regel;

 air=&ha04030 : ground=&hd4a6b4 : tekst=&h247efc

De kleurwaarden voor de lucht en de vloer zijn hier opgenomen in hexadecimale vorm, dat wil zeggen



&h blauw groen rood

waarbij ieder kleurelement een waarde van 00 (donkerst) tot FF (helderst) kan krijgen (0-9 en A-F, met twee tekens per kleurelement. Verander de **ground** (vloer) en **air** (lucht) kleurwaarden, druk daarna **F5**.

In het voorbeeld is de vloerkleur lichter gemaakt, het effect is duidelijk te zien; er ontstaat een afwijking tussen de basis vloer en de elementen op de vloer. De onderstaande elementen moeten dan ook worden bijgewerkt als je de kleur van de vloer verandert;



De waarde van **tekst** verandert de kleur van de tekst, dat wordt alleen gebruikt in het menu (plus rood – zie volgende pagina, waar direct de waarde 255 voor is gebruikt).

  

Wat ook moet worden aangepast is het menu plaatje (en daarbij ook de naam ervan in **$resource TA as "Sokoban.BMP"** zetten om het te gebruiken), hierboven drie voorbeelden waarbij de **air** en **tekst** kleuren zijn veranderd en een logo plaatje is gemaakt met die zelfde achtergrond kleur. Ook een aangepast pictogram is wenselijk om de verschillende versies uit elkaar te kunnen houden; hier is een basis pictogram gebruikt met een nummer er in verwerkt dat de kleur van de ‘lucht’ weerspiegelt.





In versie 3 zijn de doelen en de muur veranderd (gewoon omdat het kan) qua kleur, in versie 1 is de bediening anders (puzzelkeuze via cijfertoetsen) en is een 3x3 ‘toren’ element toegevoegd.



Het informatie schermpje dat verschijnt als je een level hebt uitgespeeld; in versie 1 is een tweede teller ingebouwd die het aantal stappen waarbij een kist geduwd is bijhoudt. Er is natuurlijk nog veel meer mogelijk maar voor een introductie lijken me deze voorbeelden me ruim voldoende. Ontwerp, programma en graphics door;

Ad C. Lockhorst

Lupinepad 7

3202 BT Spijkenisse

alockhorst@chello.nl